Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Relating compound toxicity to molecular structure using machine learning

Master Thesis

Robin Bosshard

October 16, 2023

Advisors: Dr. Eliza Harris, Dr. K. Arturi, Lilian Gasser

Department of Computer Science, ETH Zürich

Abstract

Abstract goes here.

Contents

Chapter 1

Introduction

intro

Chapter 2

Literature Review

2.1 Background

2.2 Context

Chapter 3

Material and Methods

3.1 Invitrodb

The most recent release of the ToxCast's (Toxicity Forecaster) database, referred to as invitroDBv3.5, serves as a source of an extensive collection of high-throughput screening (HTS) targeted bioactivity data. This database encompasses information on a total of 9541 compounds, selectively screened across 2205 assay endpoints. This resource originated from the collaboration of two prominent institutions: the United States Environmental Protection Agency (EPA) through its ToxCast program and the National Institutes of Health (NIH) via the Tox21 initiative. Incorporating data collected from diverse research laboratories, this relational database is openly accessible to the public and can be downloaded directly from the official ToxCast website.

3.1.1 Data Overview

Presence Matrix

Consider a collection of m assay endpoints, denoted by $A = \{a_1, a_2, \ldots, a_m\}$ and a set of n compounds represented as $C = \{c_1, c_2, \ldots, c_n\}$. To facilitate data comprehension, we introduce a *presence matrix* $P \in \{0,1\}^{m \times n}$. Rows, indexed by i, represent assay endpoints a_i, while columns, indexed by j, denote presence (1) or absence (0) of compound c_j in those endpoints. Matrix P is sparse due to the selective testing of compounds across different assay endpoints. A compound is considered present in an assay endpoint if it has undergone testing and a corresponding concentration-response series is available. See Figure 3.1 for a visual of the *presence matrix* P covering all assay endpoints and compounds in *invitroDBv3.5*.

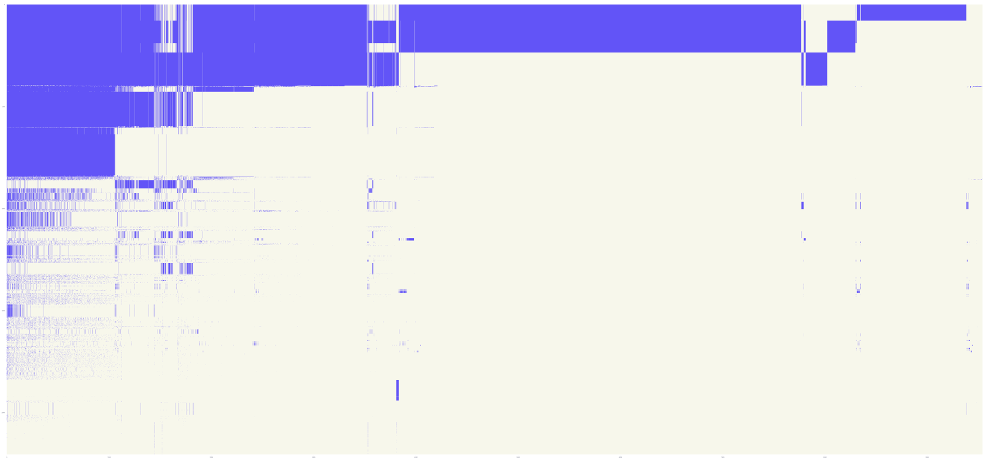

Figure 3.1: The *presence matrix* P covering all assay endpoints and compounds available in *invitroDBv3.5* with $m = 2205$ assay endpoints and $n = 9541$ compounds. The presence matrix is organized by sorting it based on the number of compounds present in each assay endpoint and the compounds are arranged in descending order of their presence frequency. The total count, where $P_{ij} = 1$, indicates the availability of $3\,342\,377$ concentration-response series for downstream analysis.

Subsetting data

We exclusively consider assay endpoints that have been tested with a minimum of 2000 compounds. This criterion ensures the availability of sufficient data for the training of a machine learning model. Refer to Figure 3.2 for a visual representation of the *presence matrix* P, which now encompasses only the resulting subset of all assay endpoints within *invitroDBv3.5*. From now on, we will call this specific subset the data that we will be focusing on for this thesis.

Concentration-Response Series

A *concentration-response series* is represented as a set of k concentration-response pairs:

$$S = \{(conc_1, resp_1), (conc_2, resp_2), \ldots, (conc_k, resp_k)\}$$

For each entry in the presence matrix P with $P_{ij} = 1$, we collect the corresponding concentration-response series S_{ij} for the compound c_j in the assay endpoint a_i. We analyse in total $\sum_{i,j} P_{ij} = 1\,372\,225$ concentration-response series, comprising a sum of $\sum_{i,j} |S_{ij}| = 48\,861\,036$ concentration-response pairs across all compounds and assay endpoints. We get the concentration-response pairs by combining tables mc0, mc1, and mc3 from invitroDBv3.5. We also gather necessary sample information such as well type, row, and

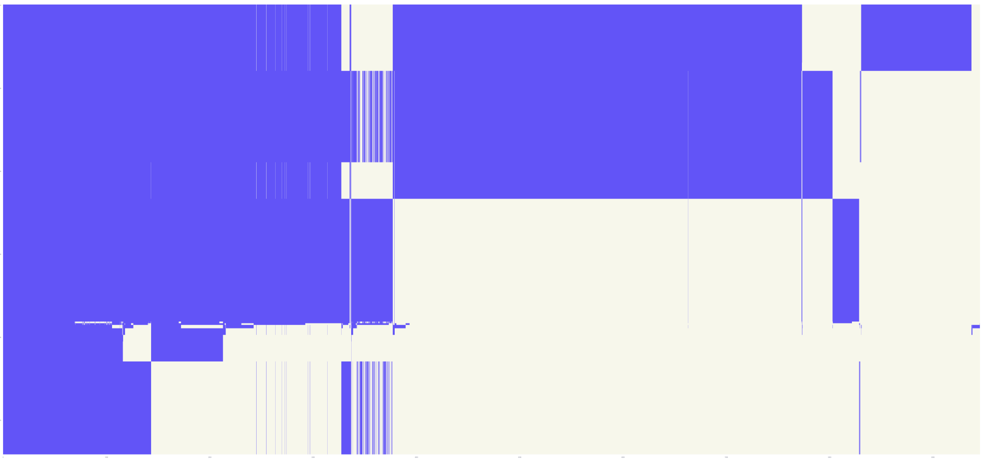

Figure 3.2: The *presence matrix* P covering only the subset of all of assay endpoints available in *invitroDBv3.5*, considered for this thesis, encompassing $m = 271$ assay endpoints and $n = 9456$ compounds. The total count, where $P_{ij} = 1$, indicates the availability of 1 372 225 concentration-response series for downstream analysis.

column index from the assay well-plate. The concentrations are transformed to the logarithmic scale using the unit μM (micromolar), while the responses are normalized to either fold-induction or percent-of-control units. Figure 3.3 showcases a single concentration-response series for some compound tested within a assay endpoint.

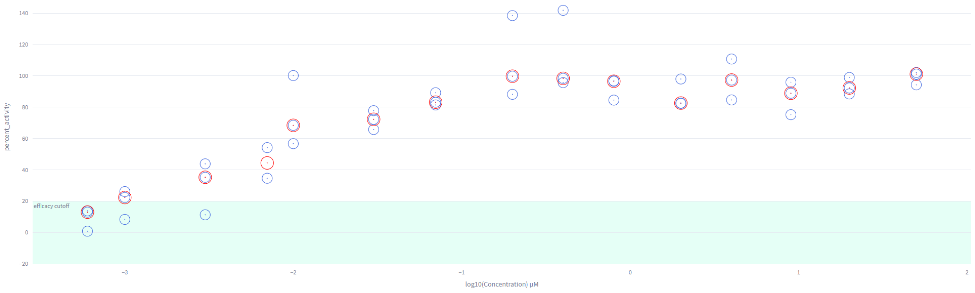

Figure 3.3: A concentration-response series for the compound *Estropipate* (DTXSID3023005) in the assay endpoint *TOX21_ERa_LUC_VM7_Agonist* (aeid=788). The series has a total of $k = 45$ concentration-response pairs and is composed of $n_{conc} = 15$ concentration groups, each with $n_{rep} = 3$ replicates.

In this section, we demonstrate the significance of variations in concentration-response pairs among different compounds and assay endpoints. In practice, concentrations are often subjected to multiple testing iterations, resulting in the formation of distinct concentration groups. Within each concentration group, the number of replicates is indicated by n_{rep}. We introduce the following quantities corresponding to a concentration-response series for a compound c_i in a given assay endpoint a_i:

- $n_{\text{datapoints}_{i,j}}$: the total number of concentration-response pairs
- $n_{\text{groups}_{i,j}}$: the number of distinct concentrations tested
- $n_{\text{replicates}_{i,j}}$: the number of replicates for each concentration group
- $min_{\text{conc}_{i,j}}$: the lowest concentration tested
- $max_{\text{conc}_{i,j}}$: the highest concentration tested

For an overview of these quantities across the entire set of considered concentration-response series, please refer to Figure 3.4. This figure illustrates the above metrics aggregated by their means, grouped by assay endpoints and compounds.

3.2 Pytcpl

We introduce pytcpl, a streamlined Python package inspired by the R package tcpl, designed for processing high-throughput screening data. The package primarily focuses on providing essential features such as concentration-response curve fitting and allows for continuous hit-calling for compound bioactivity across diverse assay endpoints, akin to tcplfit2. Invitrodb version 3.5 release can optionally serve as backend database if desired. The package optimizes data storage and provides compressed raw data and metadata from *invitroDB* in Parquet files. This efficient strategy reduces storage needs, resulting in just 4 GB within the repository—compared to the original 80 GB database. This obviates the need for a cumbersome, large-scale database installation, rendering downstream analysis more accessible and efficient. Our package is crafted to accomodate cusomizable processing steps and facilitate interactive data visualization with curve surfer. Moreover, it empowers Python-oriented researchers to seamlessly engage in data analysis and exploration.

3.2.1 Pipeline

1. Data collection
2. Cutoff determination and filtering (Meet conditions for curve fitting)
3. Curve fitting
4. Hit calling

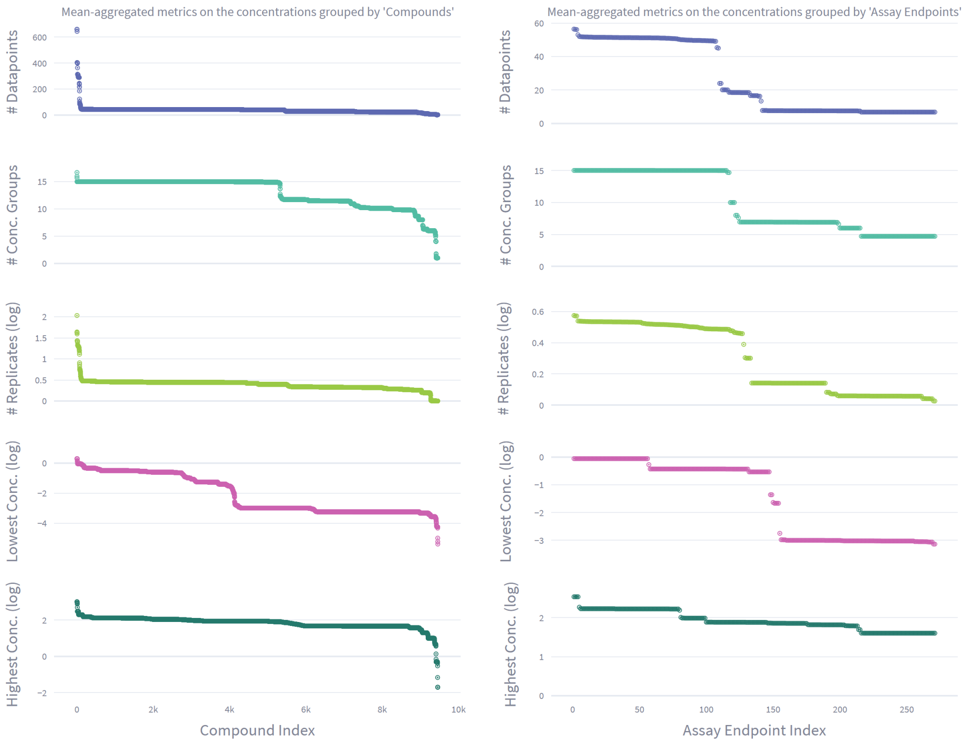

Figure 3.4: Concentration metrics aggregated by their means, grouped by compounds in the first column and assay endpoints in the second column. The first row shows the total number of concentration-response pairs, the second row shows the number of distinct concentrations tested and the third row shows the number of replicates (log-scale) for each concentration group. The fourth and fifth row show the lowest and highest concentration (log-scale) tested, respectively.

Data Collection

First, all datapoints are collected from the database and assigned to the concentration response-series belonging to the respective compound in the corresponding assay endpoint.

Curve Fitting

Introduce all candidate fit models, discuss the pros and cons of each model. Discuss the fitting procedure, how the models are fitted, Maximum Likelihood Estimation

Hit Calling

Akaike criterion, probability of being active, etc..

3.2.2 Curve Surfer

Data visualization, overview of what is possible with the tool. Filter by assay endpoint, compound, etc.

3.3 Machine Learning Pipeline

3.3.1 Preprocessing

Subselecting the columns from the output tables generated by pytcpl: DTXSID identifier and continuous hitcall value. The feature inputs to the machine learning model is a molecular structure represented as fingerprint generated from a SMILES string uniquely determined by the compounds DTXSID identifier. The SMILES string is a linear representation of a compound's molecular structure. The SMILES string is converted to a molecular graph, which is then converted to a feature vector. The feature vector is then used to train a machine learning model. The machine learning model is then used to predict the hitcall value for a given compound. The machine learning pipeline is illustrated in Figure **??**.

3.3.2 Binary Classification

The goal is to predict whether a compound is active or inactive for a given assay endpoint. We can formulate this as a binary classification problem, where the input is the compound's molecular structure fingerprint and the output is the hitcall value binarized by some decision threshold. The hitcall value is rendered to a binary variable, where 1 indicates that the compound is active and 0 indicates that the compound is inactive.

3.3.3 Regression

3.3.4 Massbank Validation

Chapter 4

Results and Discussion

sectionResults sectionEvaluation sectionDiscussion

Chapter 5

Conclusion

Appendix A

Appendix

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):

First name(s):

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

Signature(s)

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.